基于矩阵保留策略的邻域粗糙集属性约简算法 *

高 阳，刘遵仁，纪 俊

(青岛大学 计算机科学技术学院, 山东 青岛 266071)

摘 要：属性约简对于数据处理来说意义重大。在基于邻域粗糙集的属性约简算法中，正域计算是保证其有效性的重要依据，也是影响其时间开销的最主要部分。为了减少算法时间开销，通过对现有算法 FHARA 的正域计算进行改进，采取保留策略，利用矩阵保留度量计算值的平方，将原本 n 维上的计算改进为 1 维上的计算，从而缩减了每次度量计算的计算时间，并在此基础上提出了基于矩阵保留策略的邻域粗糙集属性约简算法，最后通过多个 UCI 数据集验证了该算法。与现有算法相比较，实验结果表明，对大部分数据集而言，该算法能有效且更快速地得到数据集的属性约简。

关键词：邻域粗糙集；正域；属性约简；快速算法

中图分类号：TP18 doi: 10.3969/j.issn.1001-3695.2018.05.0390

Neighborhood rough set attribute reduction algorithm based on matrix reservation strategy

Gao Yang, Liu Zunren, Ji Jun

(*College of Computer Science & Technology*, *Qingdao University*, *Qingdao Shandong* 266071, *China*)

Abstract: Attribute reduction is of great importance for data processing. For an attribute reduction algorithm based on the neighborhood rough set model, the calculation of the positive region is the necessary basis of its efficient performance and the uppermost part of its time cost. In order to reduce the time overhead of the algorithm, this paper improved the positive domain calculation of the existing algorithm FHARA, adopted the reservation strategy and used the matrix to preserved the square of the calculated values. The original n-dimensional computation was improved to 1 dimensional computation, which reduced the computation time of each metric calculation. On this basis, this paper proposed a neighborhood rough set attribute reduction algorithm based on the matrix reservation strategy. Finally, the algorithm was verified by multiple UCI data sets. Compared with existing algorithm, the experimental results show that for most data sets, the algorithm can get the attribute reduction of the dataset more effectively and quickly.

Key words: neighborhood rough set; positive region; attribute reduction; fast algorithm

chinaXiv:201810.00045v1

0 引言

随着信息技术的高速发展，人们不仅面临着数据量爆炸的问题，还有更重要的数据的高维度问题，而处理高维数据时，“维度灾难”现象十分普遍[1]。因此，属性约简对于一个数据量庞大的数据集而言是十分有意义的，可以减小维数灾难造成的影响。

粗糙集理论在数据的属性约简方面得到了广泛的应用。经典的 Pawlak 粗糙集[2]定义在经典的等价划分和等价类基础上，保证了粒度计算的进行。这种处理方式只适合处理离散型变量，而对于现实应用中广泛存在的数值型数据类型处理时，需要将数值型数据进行离散化，这种处理会改变数据原始的属性性质，造成信息损失，而离散化的方法不同又会使得处理结果不同，这严重制约了粗糙集理论的应用。为了解决这一问题，Zadeh[3]提出了信息粒化和粒度计算的概念，并给出了进行粒度计算的基本框架。Lin[4]在信息粒化、粒度的基础上提出了邻域模型的概念。胡清华等人[7]在对基本邻域信息粒子进行邻域粒化和粗糙逼近的基础上，提出了邻域信息系统和邻域决策表模型。最后，经过各方研究后提出的邻域粗糙集模型方法将经典粗糙集的等价近似与邻域逼近相结合，使之能够同时支持数据型和离散型两种数据类型，扩大了粗糙集理论的应用范围[2~8]。

但是与经典的 Pawlak 粗糙集不同，邻域粗糙集模型定义了样本间的 δ-邻域，在对邻域粗糙集的正域进行计算时，需要遍历所有样本，通过度量计算来确定样本的 δ-邻域关系，因此邻域实数空间下的计算量要比经典离散空间下的计算量大得多[9~13]，这导致了基于邻域粗糙集的属性约简算法在处理数据时往往时间开销过大的现象。

为了缩减时间开销，Hu 等人[9]提出了基于前向贪心策略的属性约简算法 F2HARNRS(fast forward heterogeneous attribute

收稿日期：2018-05-11；**修回日期：**2018-07-02 **基金项目：**国家自然科学基金资助项目（61503208）

作者简介：高阳（1994-），女，山东菏泽人，硕士研究生，主要研究方向为粗糙集理论（gy20120904@163.com）；刘遵仁（1963-），男，副教授，硕导，博士，主要研究方向为粗糙集理论、智能计算、数据挖掘等；纪俊（1982-），男，博士，主要研究方向为数据挖掘、大数据应用、转化医学等.

ChinaXiv合作期刊
chinaXiv:201810.00045v1

reduction based on neighborhood rough sets)。随后 Liu 等人[10]对该算法的正域计算进行了改进，提出了更快速的属性约简算法 FHARA(fast hash attribute reduct algorithm)，减少了 F2HARNRS 算法的正域计算时间开销。

基于以上研究，本文对 FHARA 算法进行了改进，采用矩阵保留样本间的度量计算，使得增维后只需做 1 维上的度量计算，从而减少了正域计算的计算量。通过与 FHARA 算法比较，实验证明该算法能够更快速地得到数据集的属性约简。

1　基本概念

1.1　邻域粗糙集

定义 1　给定 n 维实数空间 R^n，对于空间中的任意两个点 $x_i=(x_{i1},x_{i2}\dots,x_{in})$ 和 $x_j=(x_{j1},x_{j2},\dots,x_{jn})$，定义 $d(x_i,x_j)$ 是 R^n 上的一个度量计算，满足

$$d(x_i,x_j)=\left(\sum_{p=1}^{n}\left|x_{ip}-x_{jp}\right|^2\right)^{\frac{1}{2}}$$

定义 2[7]　对于一个给定的实数空间上的非空有限集合 $U=\{x_1,x_2\dots,x_n\}$，其中 U 为论域。对于 U 上的任意样本 x_i，定义其 δ-邻域为 $\delta(x_i)=\{x_j\mid x_j\in U,d(x_i,x_j)\le\delta\}$，其中 $\delta\ge 0$。$\delta(x_i)$ 称为由 x_i 生成的 δ 邻域信息粒子，简称为 x_i 的邻域粒子。

1.2　邻域决策系统

定义 3　对于四元组 $\mathrm{NDT}=(U,A,V,f)$，其中 U 是论域；$A=C\cup D$，C 是条件属性，D 是决策属性，且 $C\cap D=\varnothing$，$C\ne\varnothing$，$\mathrm{D}\ne\varnothing$；V 是信息函数 f 的值域；f 是 $U\times A\to V$ 的映射，那么称这个四元组为邻域决策系统。

定义 4　对于一个给定的邻域决策系统 $\mathrm{NDT}=(U,C\cup D,V,f)$，$D$ 将 U 划分为 N 个等价类：$D_1,D_2,\dots,D_N$，$\forall B\in C$，定义决策属性 D 关于 B 的下近似和上近似为

$$\underline{N_B D}=\bigcup_{i=1}^{N}\underline{N_B D_i},$$

$$\overline{N_B D}=\bigcup_{i=1}^{N}\overline{N_B D_i}$$

其中，

$$\underline{N_B D_i}=\{x_i\mid\delta_B(x_i)\subseteq D_i,x_i\in U\},$$

$$\overline{N_B D_i}=\{x_i\mid\delta_B(x_i)\cap D_i\ne\varnothing,x_i\in U\}$$

根据定义 1：

$$\delta_B(x_i)=\{x\mid d(B(x_i),B(x))\le\delta,x\in U\}$$

定义决策属性集 D 关于 B 的边界域为 $BN(D)=\overline{N_B D}-\underline{N_B D}$，正域为 $Pos_B(D)=\underline{N_B D}$。

定义 5　根据定义 4，进一步定义决策属性 D 对 B 的依赖性为：

$$\gamma_B(D)=|Pos_B(D)|/|U|$$

2　邻域粗糙集属性约简算法

对一个给定的数据集，如何设计以及利用有效的算法来删除冗余属性，寻找最小属性约简是一个 NP-Hard 问题。

定义 6[2]　给定有限集合 $B\subseteq C$，若满足 $Pos_B(D)=Pos_C(D)$，则称 B 是一个独立属性子集；如果对 $\forall a\in B$，$Pos_{B-\{a\}}(D)<Pos_B(D)$，则称 B 为 C 的一个属性约简。

2.1　F2HARNRS 算法和 FHARA 算法介绍

贪心策略能够在较少时间内求解最优解或次优解。Hu 等人[9]首先根据依赖性函数定义了条件属性对分类的贡献，称之为属性重要度，可以作为属性集合重要性的评价指标；然后根据属性重要度指标构造了一种基于邻域粗糙集的前向贪心属性约简算法，即 F2HARNRS 算法。

定义 7[9]　给定邻域决策系统 $\mathrm{NDT}=(U,C\cup D,V,f)$，$\forall B\in C$，$\forall a\in C-B$，定义属性 a 相对于集合 B 的属性重要度为

$$\mathrm{SIG}(a,B,D)=\gamma_{B\cup a}(D)-\gamma_B(D)$$

上式等价于

$$\mathrm{SIG}(a,B,D)=|Pos_{B\cup a}(D)|-|Pos_B(D)|$$

F2HARNRS[9]算法的基本思想是：初始化约简集合为空集，此时决策属性对集合的依赖度为 0，每次计算全部剩余属性的属性重要度，并从中选取重要度最大的属性，即让当前正域中样本个数增加最大的属性加入到约简集合中，直到所有剩余属性的重要度全为 0，即样本全划入当前正域中时，此时加入新的属性函数依赖值保持不变。输出集合，此时决策属性对集合的依赖度为 1，即当前正域为论域。这种算法保留了重要度最大的属性，相当于保证核不被约简。其中，当新增加属性时，原本属于正域的样本不会变为非正域样本，因此，在算法的计算过程中，每次只需对还未判定为正域的样本进行正域计算，减少了样本判断次数。

F2HARNRS 算法的正域计算可以表示为图 1。样本 x_i 需要和论域内所有样本做度量计算，其时间复杂度为 $O(m|U^2|)$。

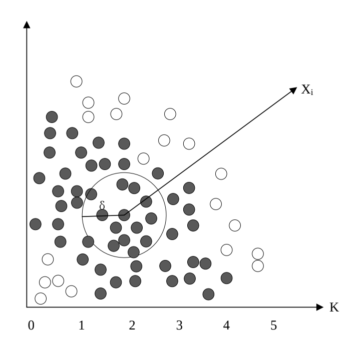


图 1　F2HARNRS 算法的正域计算

之后 Liu 等人[10]在 Hu 等人[9]的基础上改进了 F2HARNRS 算法的正域计算方法，提出了更快速的 FHARA 算法。

FHARA 算法的正域计算方法：利用映射函数 $B_k=\{x_i\mid\forall x_i\in U\wedge[f(x_0,x_i)/\delta=k]\}$ 将论域中的样本划分到有限集合 $B_0,B_1,\cdots,B_k$ 中，其中 x_0 是论域 U 中的一个特殊样本，其定义为 $x_0=\{x_0\mid\forall a\in C,a(x_0)=\min[a(x_i)]\}$，$x_i\in U$。如果样本 $x_i\in B_q$，那么 δ-邻域只存在于 $B_{q-1}\cup B_q\cup B_{q+1}$ 中。

FHARA 算法的正域计算可以表示为图 2。x_i 只需和自身所在集合以及相邻集合中的样本做度量计算，其时间复杂度为

ChinaXiv合作期刊

$O(q.n|U^2|)$, $q = 4/[\max d(x_i, x_j)/\delta]$ 。

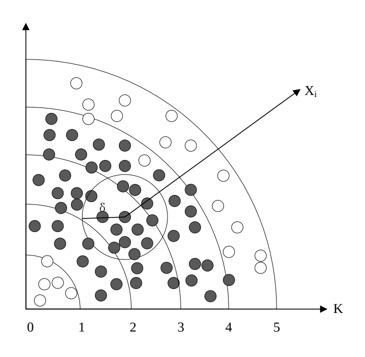


图 2　FHARA 算法的正域计算

根据以上算法分析，假设某一数据集中包含 m 个属性，$|U|$ 个样本，且约简结果中包含 k 个属性，每增加一个属性正域中增加 $\frac{|U|}{k}$ 个样本，则算法最大计算量为

$$1\cdot m|U|^2 + 2\cdot(m-1)\frac{k-1}{k}|U|^2 + \ldots + k\cdot(m-k)\frac{1}{k}|U|^2$$

2.2　基于矩阵保留策略的邻域粗糙集属性约简算法

分析上述算法的最大计算量可知，在算法计算过程中，不同维数下的度量计算是互相独立的。例如，样本 $x_i(x_{i1}, x_{i2})$ 和 $x_j(x_{j1}, x_{j2})$ 在 2 维空间上的度量计算为 $d_2(x_i,x_j) = \left((x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2\right)^{\frac{1}{2}}$，其增维后的样本 $x_i(x_{i1}, x_{i2}, x_{i3})$ 和 $x_j(x_{j1}, x_{j2}, x_{j3})$ 在 3 维空间上的度量计算为 $d_3(x_i,x_j) = \left((x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2 + (x_{i3}-x_{j3})^2\right)^{\frac{1}{2}}$。

然而这两次度量计算显然是有联系的，即 $d_3(x_i,x_j) = \left((d_2(x_i,x_j))^2 + (x_{i3}-x_{j3})^2\right)^{\frac{1}{2}}$。因此，如果在计算 $d_3(x_i,x_j)$ 前保留之前的度量计算 $d_2(x_i,x_j)$，那么增维后样本间的度量计算 $d_3(x_i,x_j)$ 只需要做 1 维上的计算，即 $(x_{i3}-x_{j3})^2$，而不必做原本 3 维上的计算。

同理，推广至 n 维，当样本增加至 n+1 维时，FHARA 算法的度量计算需要做 n+1 维上的计算，而采用保留策略将样本在 n 维空间上的度量计算保存下来，那么增至 n+1 维时只需做 1 维上的度量计算。据此，本文对 FHARR 算法的正域计算作出改进，然后提出基于矩阵保留策略的邻域粗糙集快速属性约简算法(fast attribute reduction based on matrix reservation strategy，FARBMRS)。

针对以上分析，提出以下保留策略。

改进：设当前属性约简集合 $red \in C$，在求属性 $\forall a \in C - red$ 相对于 red 的重要度之前，先在 red 下做还未判定为正域的样本与所需计算样本间的度量计算，并且用矩阵 $dist[U \times U]$ 对所求出的度量计算值的平方进行保存，那么在增维后求属性 $\forall a \in C - red$ 的相对于 red 重要度时只需从矩阵中找出相应值，再加上 1 维上的度量计算值即可。

可知，度量计算的耗时和度量计算次数主要影响着算法的时间开销。以上策略的优点在于：在度量计算次数不变的情况下，将每次度量计算过程中原本 n 维上的计算改进为 1 维上的计算，缩减了每次度量计算的计算时间，从而达到了缩减算法时间开销的目的。

改进后的正域计算 $Pos(U, a \cup D, \delta, dist)$ 如算法 1 所示。

算法 1

Input: $NDT = (U, a \cup D, V, f)$.

Output: 正域 pos .

Step 1　for each $x_i \in U$

　　$Hash(P(x_i), B_k)$;

　end for

Step 2 初始化 $pos = \varnothing$

Step 3 for each $x_i \in U$　$(x_i \in B_k)$

　　flag=0;

　　for each $x_j \in B_{k-1} \cup B_k \cup B_{k+1}$

　　　if $dist(x_i, x_j) + (a_i - a_j)^2 \le \delta^2$ && $D(x_i) \ne D(x_j)$

　　　　flag=1;

　　　　break;

　　　end if

　　end for

　　if $(\text{flag} \ne 1)$

　　　$pos \leftarrow x_i$;

　　end if

　end for

Step 4　return　pos

其中，Step 1 中的映射函数为 $B_k = \{x_i \mid \forall x_i \in U \wedge [f(x_0, x_i)/\delta = k]\}$ 。$dist(i,j)$ 表示 x_i 和 x_j 的度量计算值。根据算法 1，每次正域计算只需要做 1 维上的度量计算，即 $(a_i - a_j)^2$ 。

结合算法 1 $Pos(U, a \cup D, \delta, dist)$，下面给出 FARBMRS 算法的具体步骤，如算法 2 所示。

算法 2

Input: $NDT = (U, C \cup D, V, f)$.

Output: 属性约简 red .

Step 1 初始化 $dist[U \times U]$，$red = \varnothing$，待检验样本 $smp_chk = U$，当前正域 $max_pos = \varnothing$，重要度最大属性 $max_i = \varnothing$

Step 2　while　$smp_chk \ne \varnothing$

　　$max_pos = \varnothing$;

for each $k_i \in (C - red)$

$Pos_i = Pos(smp_chk, k_i \cup D, \delta, dist)$;

if　$|max_pos| < |Pos_i|$

　　$max_pos = Pos_i$;

　　$max_i = k_i$;

end if;

end for

　　if　$max_pos \ne \varnothing$

chinaXiv:201810.00045v1

ChinaXiv合作期刊

```
        red = red ∪ max_i ;
        smp_chk = smp_chk - max_pos
        dist = dist_red ;
    else
      break;
    end if
  end while
  Step 3  return  red
```

在算法2中，$dist_{red}$ 的更新公式为 $dist_{red}(i,j)=dist_{red-a}(i,j)+(a_i-a_j)^2$。

在该算法下，假设某一数据集中包含 m 个属性，$|U|$ 个样本，且约简结果中包含 k 个属性，每增加一个属性正域中增加 $\frac{|U|}{k}$ 个样本，则算法最大计算量为

$$1\cdot m|U|^2+1\cdot(m-1)\frac{k-1}{k}|U|^2+\ldots+1\cdot(m-k)\frac{1}{k}|U|^2$$

$$+1\cdot m|U|^2+1\cdot\frac{k-1}{k}|U|^2+\ldots+1\cdot\frac{1}{k}|U|^2$$

通过以上对 FARBMRS 算法的最大计算量的分析可知，在约简集合 red 增维的过程中，FARMRS 算法的正域计算每次仅需做 1 维上的度量计算，以及增加度量计算前的 $dist_{red}$ 计算。总的来说，FARBMRS 算法增加的计算量小于减少的计算量，而 FHARR 算法每次的正域计算都需要进行 n 维的计算，因此 FARBMRS 算法计算量少于 FHARR 算法的计算量，在理论上减少了时间开销。

3 实验分析

在实验部分，首先对 FARBMRS 和 FHARA 算法在时间开销的差别作出了对比，验证了算法的有效性；然后对 FARBMRS 算法相对于 FHARA 算法的效率作出分析。

3.1 实验环境

UCI（University of California Irvine）提供了一系列用于测试的标准数据集。为了验证 FARBMRS 算法的有效性，从 UCI 数据集中选取了八个具有代表性的数据集作为实验数据，描述如表 1 所示。

表 1 数据集描述

编号	数据集	样本数	属性数	类别数
1	Wine	178	13	3
2	Ionosphere	351	34	2
3	Libras movement	360	90	15
4	WDBC	569	30	2
5	Credit Approval	690	14	2
6	German Credit	1000	19	2
7	Biodeg	1055	41	2
8	Segmentation	2310	19	7

本次实验在一台 Intel(R) Core(TM) i5 CPU 和 4 GB 内存的 PC 机上，采用 Windows 10 环境下的 MATLAB R2016b 进行算法仿真。

3.2 δ 的取值

在计算各邻域样本时，δ 是一个关键的参数，它的取值直接影响着属性约简的结果。对于某一条件属性集合而言，如果 δ 的取值太大，会导致大部分样本划分在同一邻域内，使得最后得到的约简属性偏少；如果 δ 的取值太小，会使得最后约简属性偏多。δ 取值偏大或偏小得到的属性约简都不甚理想。

在邻域粗糙集中，δ 一般采用点值式的取值方法[9~18]，这对于不同数据集和分类器来说有不同的效果。本文采用文献[12]中提出的采用标准差度量 δ 的取值。标准差是数据平均值分散程度的一种度量。一个较小的标准差代表大部分数据都接近平均值，此时需要为邻域设定较小的 δ 值；反之，而一个较大的标准差需要设定较大的 δ 值。首先取每一列属性值的标准差，再将这些标准差取标准差作为 δ 值，即假设条件属性 $C=\{C_1,C_2,\ldots,C_m\}$，那么 δ 的取值公式为 $\delta=\sigma(\sigma(C_1),\sigma(C_2),\ldots,\sigma(C_m))$。大部分分类器在这种 δ 取值下可以获得良好的分类性能，分类效果较为理想。

3.3 实验结果

3.3.1 FARBMRS 算法的有效性

为了去掉量纲对数据的影响，先对样本数据进行归一化处理。根据 3.2 节中的分析，本次实验取 $\delta=\sigma(\sigma(C_1),\sigma(C_2),\ldots,\sigma(C_m))$，将 FARBMRS 和 FHARA 算法各执行 10 次，统计各自的属性约简和运行时间，并取 10 次中的最小值作为最后的运行时间。

两种算法得到的属性约简如表 2 所示。

表 2 算法得到的属性约简

数据集	FHARA 算法	FARBMRS 算法
Wine	13,10,7,5,11,1	13,10,7,5,11,1
Ionosphere	3,31,24,16,4	3,31,24,16,4
Libras movement	63,72,17,40	63,72,17,40
WDBC	23,22,28,9,25	23,22,28,9,25
Credit Approval	14,7,2,3,6,5,8,9,11,4,1,12,13	14,7,2,3,6,5,8,9,11,4,1,12,13
German Credit	2,4,10,3,6,1,9,7,8,11,5	2,4,10,3,6,1,9,7,8,11,5
Biodeg	7,3,38,37,1,14,35,8,13,10,9,12,2,31,22,33	7,3,38,37,1,14,35,8,13,10,9,12,2,31,22,33
Segmentation	19,16,11,13,17,14,1,18,2,6,5,4,8,10,12,15,7	19,16,11,13,17,14,1,18,2,6,5,4,8,10,12,15,7

根据表 2 可知，在 δ 取值相同的情况下，FARBMRS 和 FHARA 算法得到的约简结果是一样的，这证明 FARBMRS 算法是有效可行的。

chinaXiv:201810.00045v1

ChinaXiv合作期刊
chinaXiv:201810.00045v1

两种算法的运行时间如表 3 所示。

表 3　算法的运行时间/s

编号	数据集	FHARA 算法	FARBMRS 算法
1	Wine	1.031937	0.487075
2	Ionosphere	6.846643	2.375099
3	Libras movement	8.685433	5.426591
4	WDBC	9.168734	3.788283
5	Credit Approval	29.501807	6.900268
6	German Credit	87.749297	18.505205
7	Biodeg	117.338324	31.700159
8	Segmentation	380.319438	85.935464

表 3 的折线图如图 3 所示。

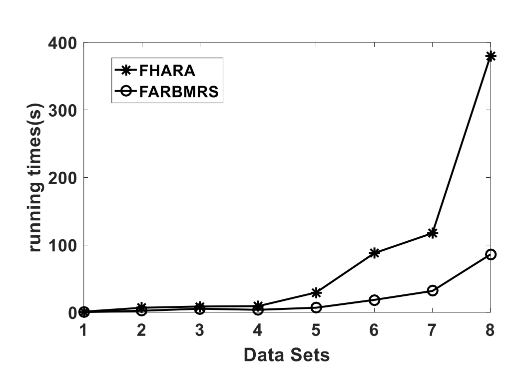


图 3　两种算法的运行时间折线图

分析图 3 运行时间折线图可以看出，FARBMRS 算法的折线一直位于 FHARR 算法折线的下方，这表示 FARBMRS 算法运行时间小于 FHARR 算法的运行时间，具有更短的时间开销。

在实验中统计了两种算法的度量计算次数，如表 4 所示。

表 4　算法的度量计算次数

编号	数据集	FHARA 算法	FARBMRS 算法
1	Wine	225680	225680
2	Ionosphere	1970004	1970004
3	Libras movement	1111901	1111901
4	WDBC	2308687	2308687
5	Credit Approval	8410959	8410959
6	German Credit	27515132	27515132
7	Biodeg	29854308	29854308
8	Segmentation	99679854	99679854

根据表 4 可知，算法的度量计算次数没有发生变化。

以上实验验证了本文 2.2 节中对两种算法计算量以及时间开销的分析。

3.3.2 FARBMRS 算法的效率

针对各数据集，用 FARBMRS 算法的运行时间与 FHARR 算法的运行时间的比值作为 FARBMRS 算法相对于 FHARR 算法的效率。其中，比值越低，说明 FARBMRS 算法的约简效率越高。

两种算法运行时间的比值如表 5 所示。

表 5　FARBMRS 算法的效率/%

编号	数据集	比值
1	Wine	47.20071
2	Ionosphere	34.68997
3	Libras movement	62.47922
4	WDBC	41.31740
5	Credit Approval	23.38930
6	German Credit	21.08872
7	Biodeg	27.01603
8	Segmentation	22.59560

表 5 的折线图如图 4 所示。

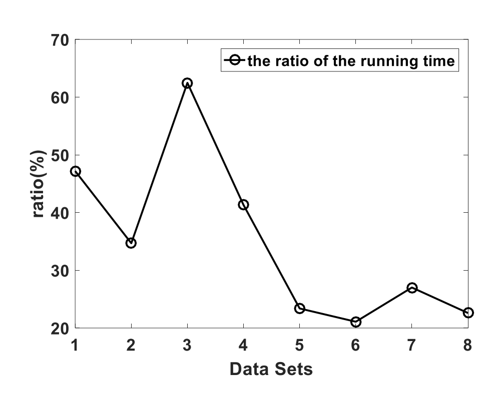


图 4　FARBMRS 算法的效率折线图

分析图 4 算法效率折线图可以看出，折线波动较大，取值区间位于 20%~65%，跨度较大。其中，大部分点的取值较低，这表明对于大部分数据集来说，FARBMRS 算法的效率较高。这种波动性与两种算法性质有关，因为在正域计算中，如果判定当前样本属于正域，则立即跳出当前循环，这说明数据集中样本在样本空间中分布会影响两种算法的约简效果。相较于 FHARA 算法，当 FARBMRS 算法增加的 $dist_{red}$ 计算量接近减少的度量计算量时，FARBMRS 算法效率较低；但是对于大部分数据集而言，FARBMRS 算法的效率较好。

4　结束语

本文对当前邻域粗糙集中的经典属性约简算法作分析，针对算法对时间复杂度的要求，对其中的 FHARA 算法的正域计算作出了改进，提出了基于矩阵保留策略的邻域粗糙集属性约简算法 FARBMRS，减少了算法时间开销，更快速地求得数据集的属性约简，且通过多个 UCI 标准数据集的实验验证，该算法是有效且更快速的。本文对 FHARA 算法的正域改进还可以与 Lou[11]的研究相结合，进一步减少算法的时间开销。

参考文献：

[1] 贺玲，蔡益朝，杨征. 高维数据聚类方法综述 [J]. 计算机应用研究, 2010, 27 (1): 23-26. (He Ling, Cai Yichao, Yang Zheng. Survey of lustering algorithms for high-dimensional data [J]. Application Research of Computers, 2010, 27 (1): 23-26.)

ChinaXiv合作期刊
chinaXiv:201810.00045v1

[2] Pawlak Z, So-Winski R. Rough set approach to multi-attribute decision analysis [J]. European Journal of Operational Research, 1994, 72 (3): 443-459.

[3] Zadeh L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic [J]. Fuzzy Sets and Systems, 1997, 90 (90): 111-127.

[4] Lin T Y. Granular Computing on binary relations I: data mining and neighborhood systems [J]. Rough Sets in Knowledge Discovery, 1998, 18 (1): 107-121

[5] 王国胤. Rough 集理论与知识获取 [M]. 西安: 西安交通大学出版社, 2001: 147-156. (Wang Guoyin. Rough set theory and knowledge acquisition [M]. Xi’an: Xi’an Jiaotong University Press, 200l: 147-156.)

[6] 胡清华，赵辉，于达仁. 基于粗糙集的符号与数值属性的快速约简算法 [J]. 模式识别与人工智能, 2008, 21 (6): 730-738. (Hu Qinghua, Zhao Hui, Yu Daren. Efficient symbolic and numerical attribute reduction with rough sets [J]. Pattern Recognition and Artificial Intelligence, 2008, 21 (6): 730-738.)

[7] 胡清华，于达仁，谢宗霞. 基于邻域粒化和粗糙逼近的数值属性约简 [J]. 软件学报, 2008, 19 (3): 640-649. (Hu Qinghua, Yu Daren, Xie Zongxai. Numerical attribute reduction based on neighborhood granulation and rough approximation [J]. Journal of Software, 2008, 19 (3): 640-649.)

[8] 胡清华，于达人. 应用粗糙计算 [M]. 北京: 科学出版社, 2012. (Hu Qinghua, Yu Daren. Applied rough sets [M]. Beijing: Science Press, 2012.)

[9] Hu Qinghua, Yu Daren, Liu Jinfu, *et al.* Neighborhood rough set based heterogeneous feature subset selection [J]. Information Sciences, 2008, 178 (18): 3577–3594.

[10] Liu Yong, Huang Wenliang, Jiang Yunliang, *et al.* Quick attribute reduct algorithm for neighborhood rough set model [J]. Information Sciences, 2014, 271 (7): 65-81.

[11] 娄畅，刘遵仁，郭功振. 基于块集的邻域粗糙集的快速约简算法 [J]. 计算机科学, 2014, 41 (S2): 337-339. (Lou Chang, Liu Zunren, Guo Gongzhen. Quick attribute reduct algorithm on neighborhood rough set based on block set [J]. Computer Science, 2014, 41 (S2): 337-339.)

[12] 刘遵仁，吴耿峰. 基于邻域粗糙集模型的高维数据集快速约简算法 [J]. 计算机科学, 2012, 39 (10): 268-271. (Liu Zunren, Wu Gengfeng. Quick reduction algorithm for high-dimensional data sets based on neighborhood rough set model [J]. Computer Science, 2012, 39 (10): 268-271.)

[13] Guo Gongzhen, Liu Zunren, Lou Chang, *et al.* Improving on a rapid attribute reduction algorithm based on neighborhood rough sets [C]// Proc of International Conference on Fuzzy Systems and Knowledge Discovery. 2016: 236-240.

[14] Chen Hongmei, Li Tianrui, Luo Chuan, *et al.* Dominance-based neighborhood rough sets and its attribute reduction [M]// Rough Sets and Knowledge Technology. [S. l.] : Springer International Publishing, 2015.

[15] Wang Changzhong, Shao Mingwen, He Qiang, *et al.* Feature subset selection based on fuzzy neighborhood rough sets [J]. Knowledge-Based Systems, 2016, 111: 173-179.

[16] 段洁，胡清华，张灵均，等. 基于邻域粗糙集的多标记分类特征选择算法 [J]. 计算机研究与发展, 2015, 52 (1): 56-65. (Duan Jie, Hu Qinghua, Zhang Lingjun, *et al.* Feature selection for muli-label classification based on neighborhood rough sets [J]. Journal of Computer Research and Development, 2015, 52 (1): 56-65.)

[17] Chen Yumin, Zeng Zhiqiang, Lu Junwen. Neighborhood rough set reduction with fish swarm algorithm [J]. Soft Computing, 2016, 21 (23): 6907-6918.

[18] Fan Anjing, Zhao Hong, Zhu William. Test-cost-sensitive attribute reduction on heterogeneous data for adaptive neighborhood model [J]. Soft Computing, 2015, 20 (12): 4813-4824.